

Project Report



SmartDoctor

Skin Cancer Detector

Theme By : - Virtusa Technologies

By:-

MSG Coders Team

Manisha Singh

Sameer Kumar

Gourav Kumar

About

SmartDoctor is an online doctor consulting service, which provides the online medical assistance in the skin related diseases. It saves the precious time of customers by finding theirs solutions to the problem in just some couple of minutes.

Having any skin related issue, fearing it might be cancer, having very tight schedule, or you do not have any idea about whom to consult or how to tackle?

Different problem , one solution-
SmartDoctor



Who we are

SmartDoctor is an online system specially designed to assist the customer in the skin related diseases (skin cancer). This system predict the type of the skin cancer he or she is suffering from and provide the required medication and precautionary tips. Through this, one can easily make an appointment to the doctor and seek for the one to one meeting. SmartDoctor has large no of doctor panel connected to it, which is ready to assist the patient any time.

The complete architecture of the smartdoctor depends upon artificial intelligence. It uses the artificial intelligence and machine learning power to predict the diseases. The user only has to upload the picture of their infected skin area rest work is done by our highly trained model working behind.

Things that make smart doctor different from the other machine-learning project is its integrations. It is integrated with the latest technologies of the market namely:- GitHub, Jenkins , docker, kubernetes, machine learning. With these powerful integrations, smartdoctor is really unique and different from others.

Currently smartdoctor is dealing with 9 types of the skin cancer diseases,

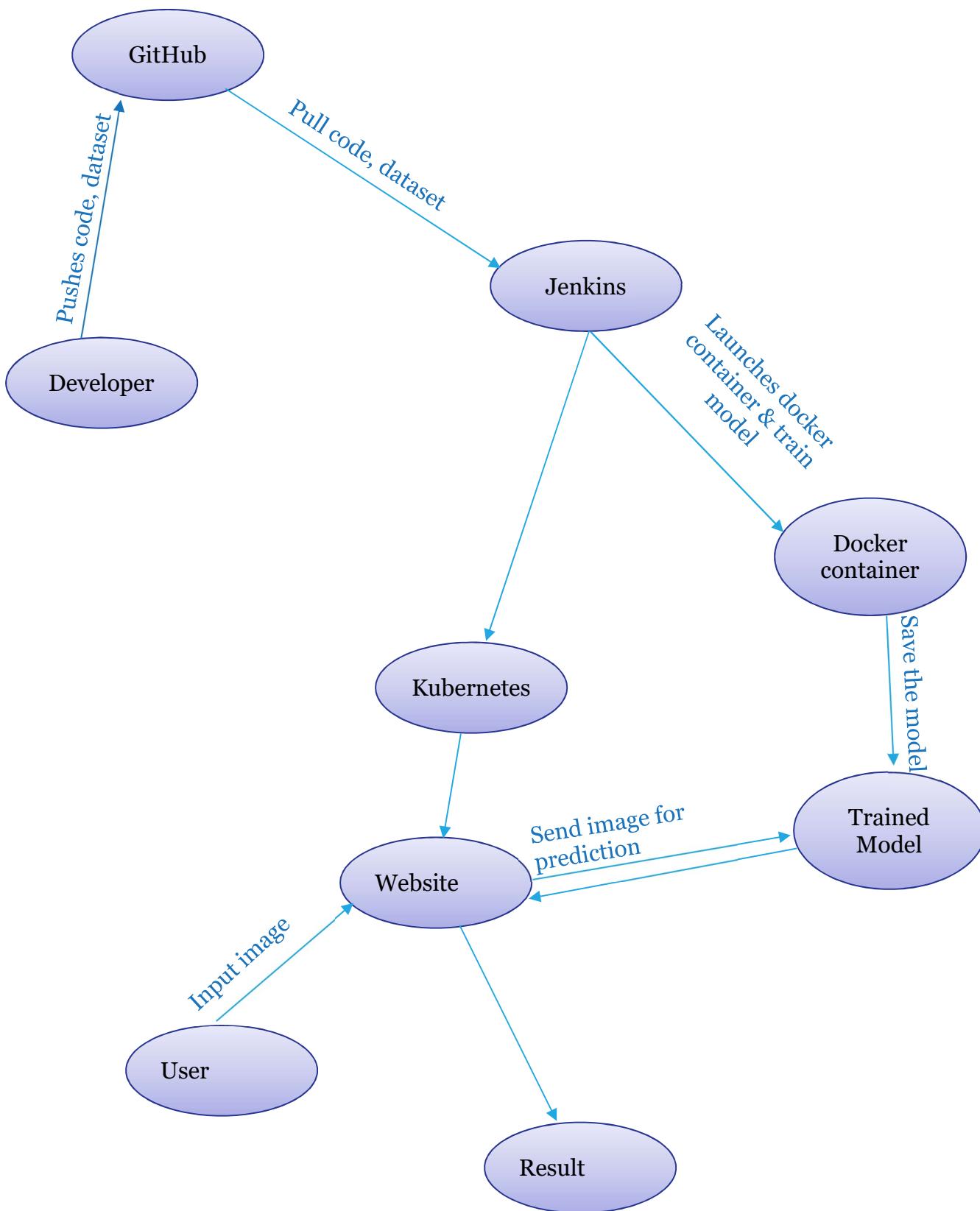
- pigmented benign keratosis
- Actinic keratosis
- Basal cell carcinoma
- Dermatofibroma
- Melanoma
- Nevus
- Seborrheic keratosis
- Squamous cell carcinoma
- Vascular lesion

SmartDoctor easily detects the above nine diseases and suggest their medicines. It is powered with a simple but powerful webui through which user can interact with the smartdoctor and enjoys its service.

Technologies used

- **Artificial Intelligence and machine learning**
 - Keras
 - Tensorflow
 - OpenCV
 - Imagenet
 - Transfer Learning
 - CNN
- **Dev-Ops**
 - Jenkins
 - GitHub
 - Git
 - Kubernetes
 - Docker
- **Web Development**
 - Html
 - JavaScript
 - Php
 - CSS
 - Bootstrap
- **Database**
 - Mysql
- **Python**
 - Datetime
 - FPDF
 - CV2 (opencv)

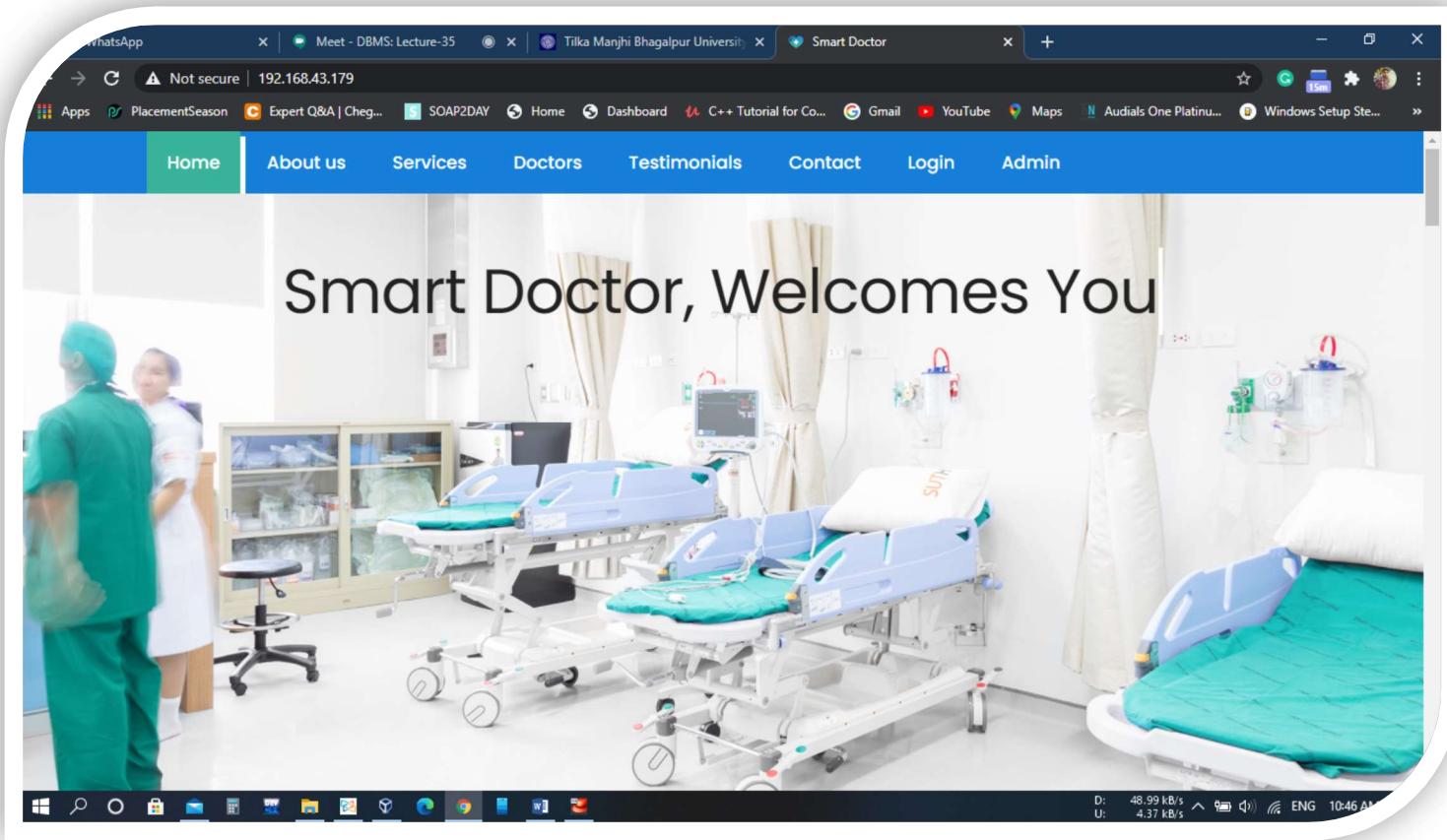
Blueprint (Architecture of SmartDoctor)



Working

SmartDoctor comes with the beautiful website to access it. User first have to get himself registered with valid email address to use the service. After the successful registration user can get into the site by login with registered email address. After login, a user portal opens where user can upload the image of his disease and has to wait 3 minutes to see the result. Generated result is in the pdf format, which consist of precautionary tips and medicine.

Along with prediction, user can also make an appointment to the concerned doctor and make a query.



Home Page of SmartDoctor website

The screenshot shows a web browser window with two tabs open:

- Registration:** This page contains fields for entering name, email, password, gender (Male), age, and city. It also includes dropdown menus for 'You are:' and 'How old are you?'. A green 'Submit' button is at the bottom.
- Login:** This page has fields for Email id and Password. Below these are buttons for 'Login As:' (User or Doctor) and a 'Logout' link.

The browser's address bar shows the URL `192.168.43.179/login.php`. The system tray at the bottom indicates network speed (D: 38.34 kB/s, U: 4.73 kB/s), battery level (ENG 10:47 AM), and signal strength.

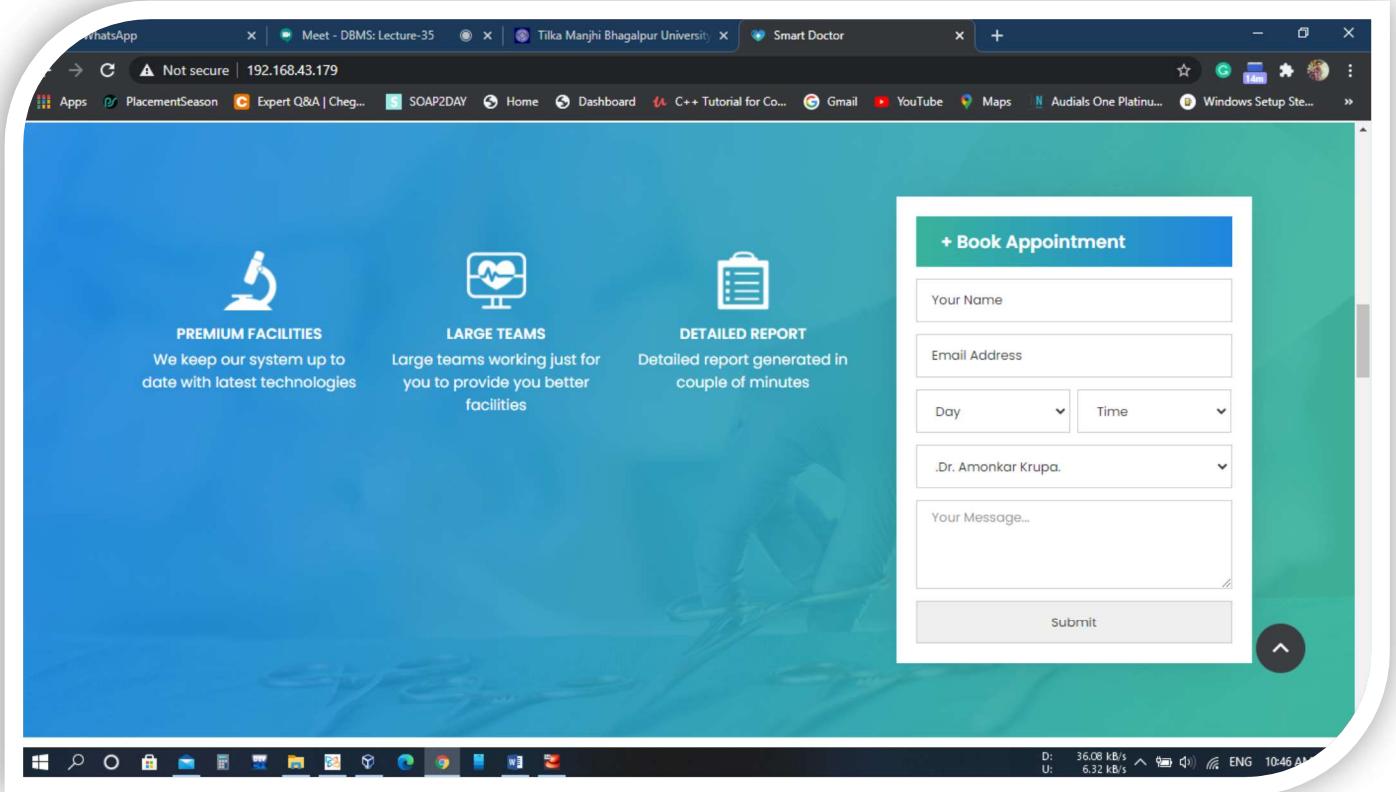
User can either login or get himself registered

The screenshot shows a web browser window with one tab open:

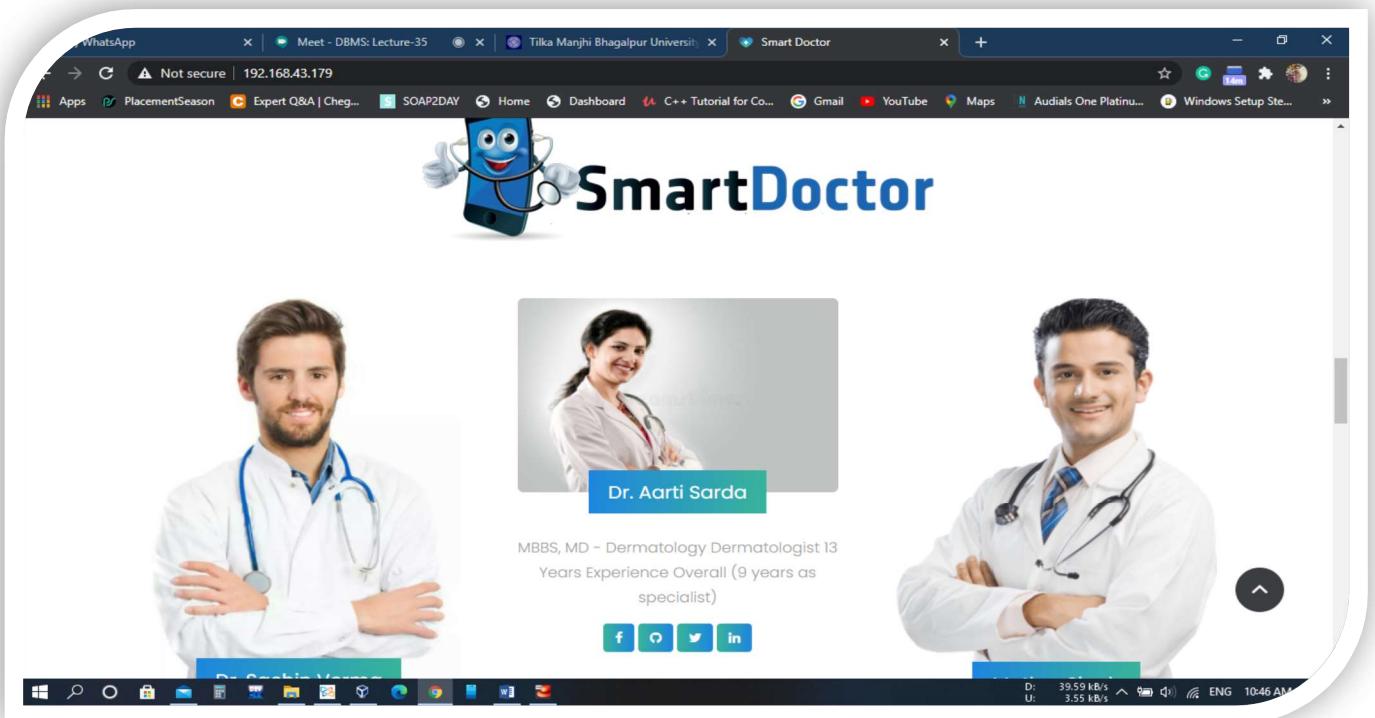
- Welcome You...**: The main heading on the page.
- Upload File**: A section with a 'Choose File' button and a note: "Note : Upload image of a skin disease." A green 'Upload' button is below the note.
- Help Center**: A section with input fields for Name, E-mail, and Message.

The browser's address bar shows the URL `192.168.43.179/patient.php`. The system tray at the bottom indicates network speed (D: 36.88 kB/s, U: 3.92 kB/s), battery level (ENG 10:47 AM), and signal strength.

User panel after successful login



Appointment booking system



Panel of doctors attached to us

Backend Working (Technical Stuffs)

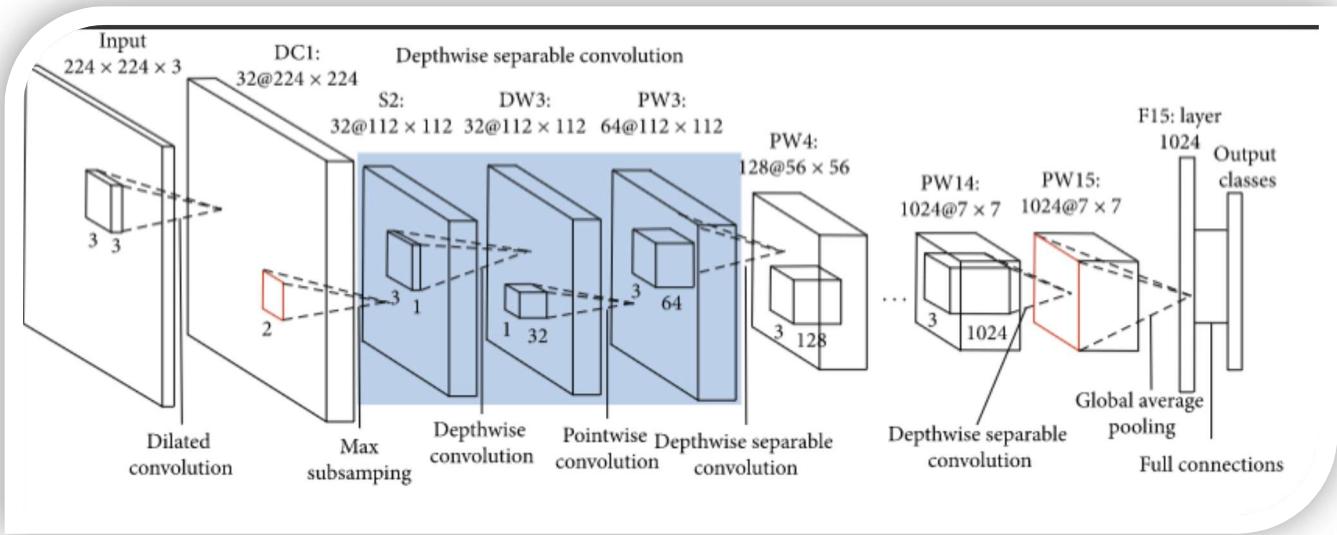
SmartDoctor uses various powerful technologies. Each technology interact with others to make it a successful and well working projects.

Various technologies are:-

- Transfer Learning along with CNN – for model creation
- Kubernetes – For deploying smartdoctor website
- Docker – model training and prediction
- Jenkins – Controlling all the activity , making complete things automatic
- Github – stores all the files and datas
- Mysql- database for website

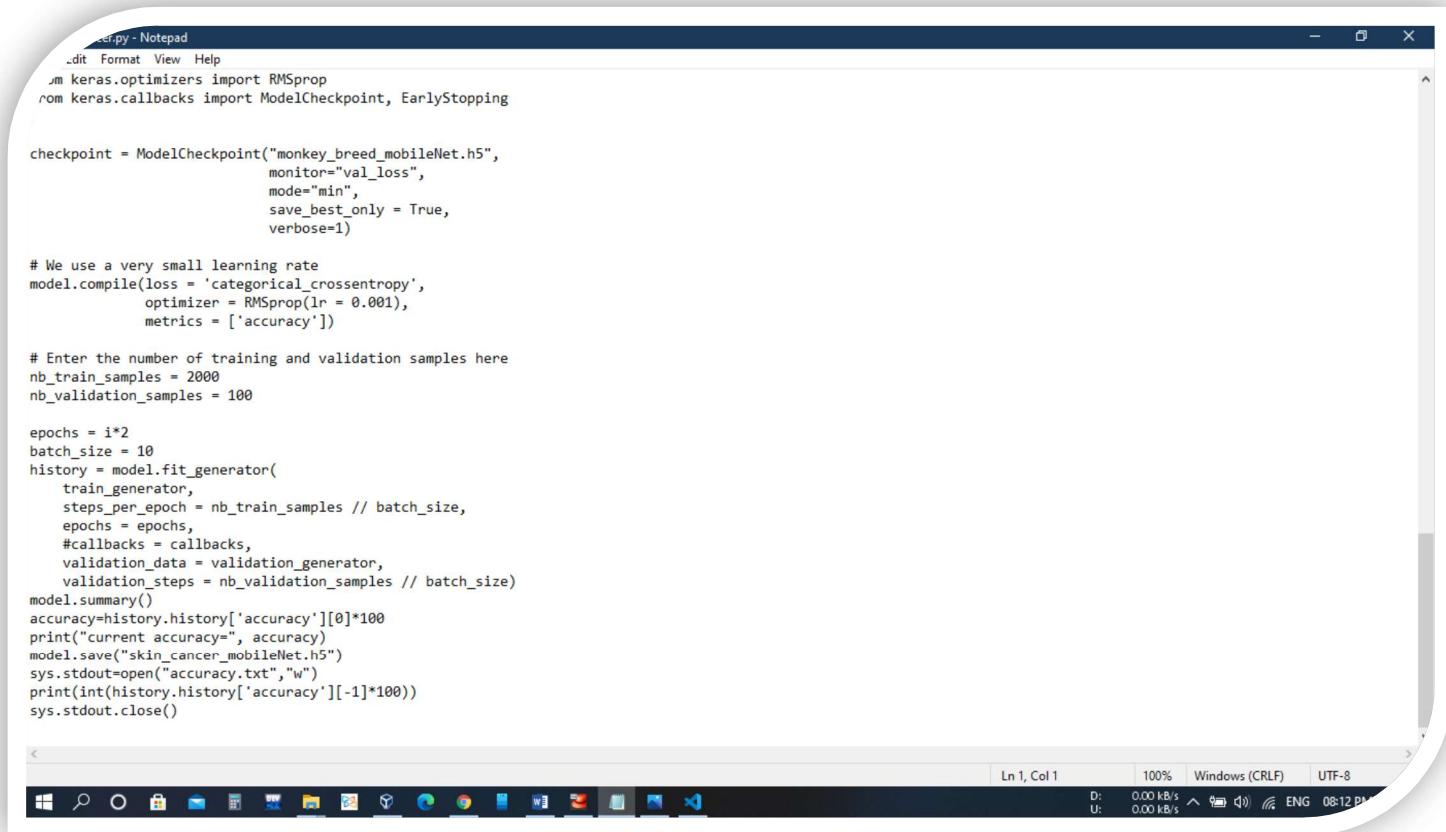
Model Creation : - Model is very important in machine learning project. For the success of the any machine learning project , model creation is the important aspect. The process of training an ML model involves providing an ML algorithm (that is, the *learning algorithm*) with training data to learn from. The term *ML model* refers to the model artifact that is created by the training process.

Our project smartdoctor is also a machine-learning project. Hence, model creation is most important part. Our model is built on the top of the mobileNet neural network. MobileNet is a streamlined architecture that uses depth wise separable convolutions to construct lightweight deep convolutional neural networks. MobileNet has fewer parameters and higher classification accuracy. SmartDoctor uses the imagenet as a weight. The complete model training is done using the transfer learning techniques. After the successful training of the model, the trained model is saved at a given location, which is used further to make the prediction.



MobileNet Architecture

The following is the snapshot of code of smartdoctor model



```
scr.py - Notepad
Edit Format View Help

from keras.optimizers import RMSprop
from keras.callbacks import ModelCheckpoint, EarlyStopping

checkpoint = ModelCheckpoint("monkey_breed_mobileNet.h5",
                             monitor="val_loss",
                             mode="min",
                             save_best_only = True,
                             verbose=1)

# We use a very small learning rate
model.compile(loss = 'categorical_crossentropy',
              optimizer = RMSprop(lr = 0.001),
              metrics = ['accuracy'])

# Enter the number of training and validation samples here
nb_train_samples = 2000
nb_validation_samples = 100

epochs = i*2
batch_size = 10
history = model.fit_generator(
    train_generator,
    steps_per_epoch = nb_train_samples // batch_size,
    epochs = epochs,
    callbacks = callbacks,
    validation_data = validation_generator,
    validation_steps = nb_validation_samples // batch_size)
model.summary()
accuracy=history.history['accuracy'][0]*100
print("current accuracy=", accuracy)
model.save("skin_cancer_mobileNet.h5")
sys.stdout=open("accuracy.txt","w")
print(int(history.history['accuracy'][-1]*100))
sys.stdout.close()

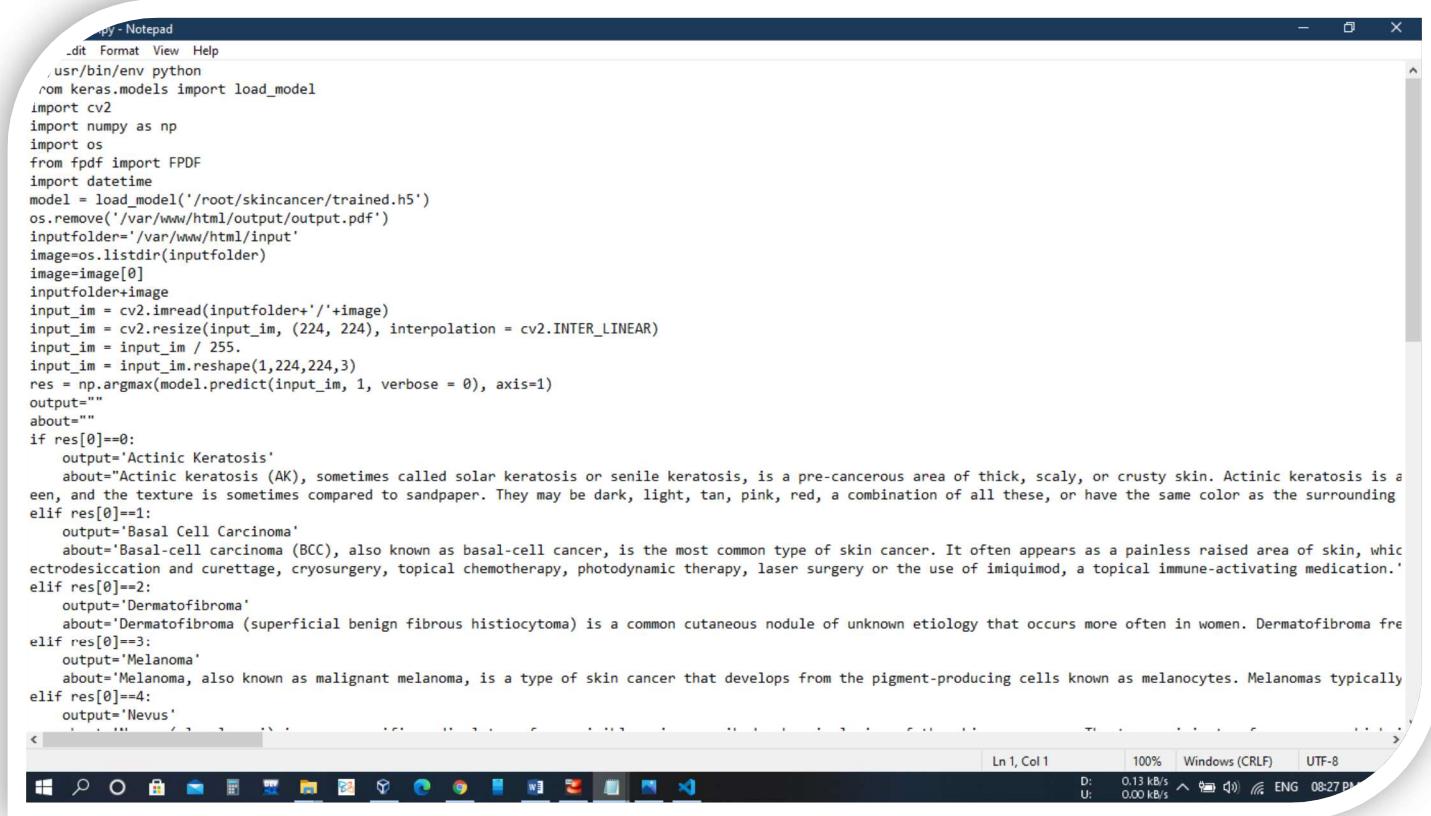
Ln 1, Col 1 100% Windows (CRLF) UTF-8
D: 0.00 kB/s U: 0.00 kB/s ENG 08:12 PM
```

Python Code of machine learning

Prediction by Trained Model : - Prediction refers as finding most appropriate answer based on the knowledge gained by the model during the training. A python script is written to read the input from the folder , process it and store the output in the output folder.

The complete process of the prediction is a two step process. In the first step it input the image from the user, given through the website, process it and predict the most appropriate result. In the second step, with the help of predicted result , a beautiful pdf file is created which consist of the predicted result , precautionary tips and medicines. This pdf file is returned as the output of the prediction process.

The following is the snapshot of the prediction code



A screenshot of a Windows Notepad window titled "py - Notepad". The code is a Python script for predicting skin cancer types based on images. It uses cv2 for image processing, keras.models for loading a trained model, and FPDF for generating a PDF output. The script reads images from a folder, processes them, and uses a pre-trained model to predict the cancer type. It then prints a detailed description of each type. The Notepad window has a standard Windows title bar and status bar at the bottom.

```
py - Notepad
File Edit Format View Help
/usr/bin/env python
from keras.models import load_model
import cv2
import numpy as np
import os
from fpdf import FPDF
import datetime
model = load_model('/root/skincancer/trained.h5')
os.remove('/var/www/html/output/output.pdf')
inputfolder='/var/www/html/input'
image=os.listdir(inputfolder)
image=image[0]
image+=image[0]
inputfolder+=image
input_im = cv2.imread(inputfolder+'/'+image)
input_im = cv2.resize(input_im, (224, 224), interpolation = cv2.INTER_LINEAR)
input_im = input_im / 255.
input_im = input_im.reshape(1,224,224,3)
res = np.argmax(model.predict(input_im, 1, verbose = 0), axis=1)
output=""
about=""
about=""
if res[0]==0:
    output='Actinic Keratosis'
    about="Actinic keratosis (AK), sometimes called solar keratosis or senile keratosis, is a pre-cancerous area of thick, scaly, or crusty skin. Actinic keratosis is a
een, and the texture is sometimes compared to sandpaper. They may be dark, light, tan, pink, red, a combination of all these, or have the same color as the surrounding
elif res[0]==1:
    output='Basal Cell Carcinoma'
    about="Basal-cell carcinoma (BCC), also known as basal-cell cancer, is the most common type of skin cancer. It often appears as a painless raised area of skin, which
ectrodesiccation and curettage, cryosurgery, topical chemotherapy, photodynamic therapy, laser surgery or the use of imiquimod, a topical immune-activating medication."
elif res[0]==2:
    output='Dermatofibroma'
    about="Dermatofibroma (superficial benign fibrous histiocytoma) is a common cutaneous nodule of unknown etiology that occurs more often in women. Dermatofibroma fre
elif res[0]==3:
    output='Melanoma'
    about='Melanoma, also known as malignant melanoma, is a type of skin cancer that develops from the pigment-producing cells known as melanocytes. Melanomas typically
elif res[0]==4:
    output='Nevus'
```

Python Code of prediction

Medical Report

2020-09-24 19:37:43.476289

Dear Sir,

According to our prediction , you are suffering from Pigmented Benign Keratosis

About the Disease

Pigmented actinic keratosis is a variant of actinic keratosis that occurs less commonly. In contrast to presenting as an erythematous plaque, its appearance can mimic a pigmented lesion (lentigo maligna or solar lentigo) or a keratinocytic lesion (lichen planus-like keratosis).

Our Suggestion for you

Seborrheic keratosis

Therapy and surgeries:-

- * Cryosurgery
- * Electrocautery
- * Curettage
- * Ablation
- * cryotherapy

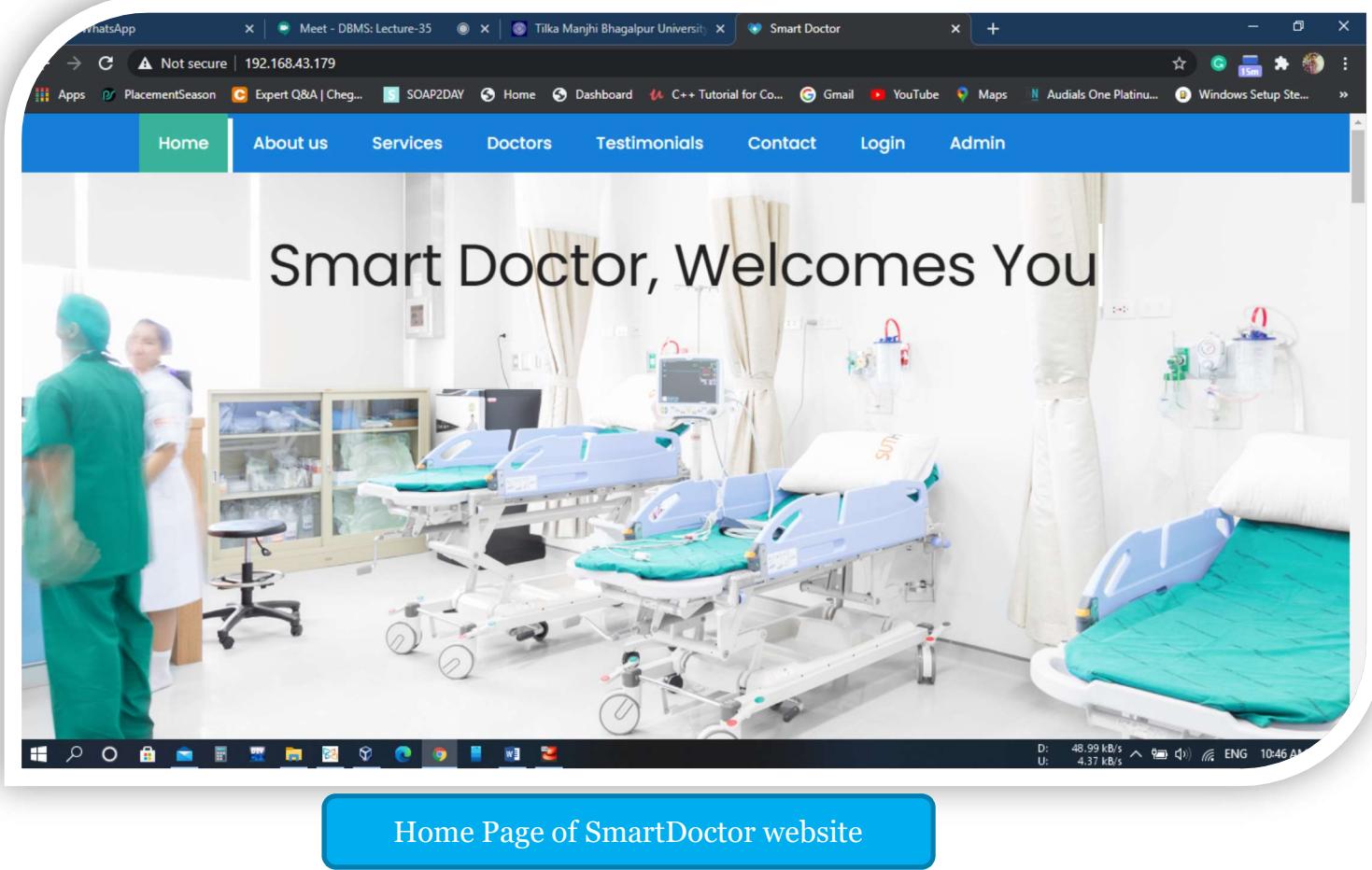
Warning Sign:-

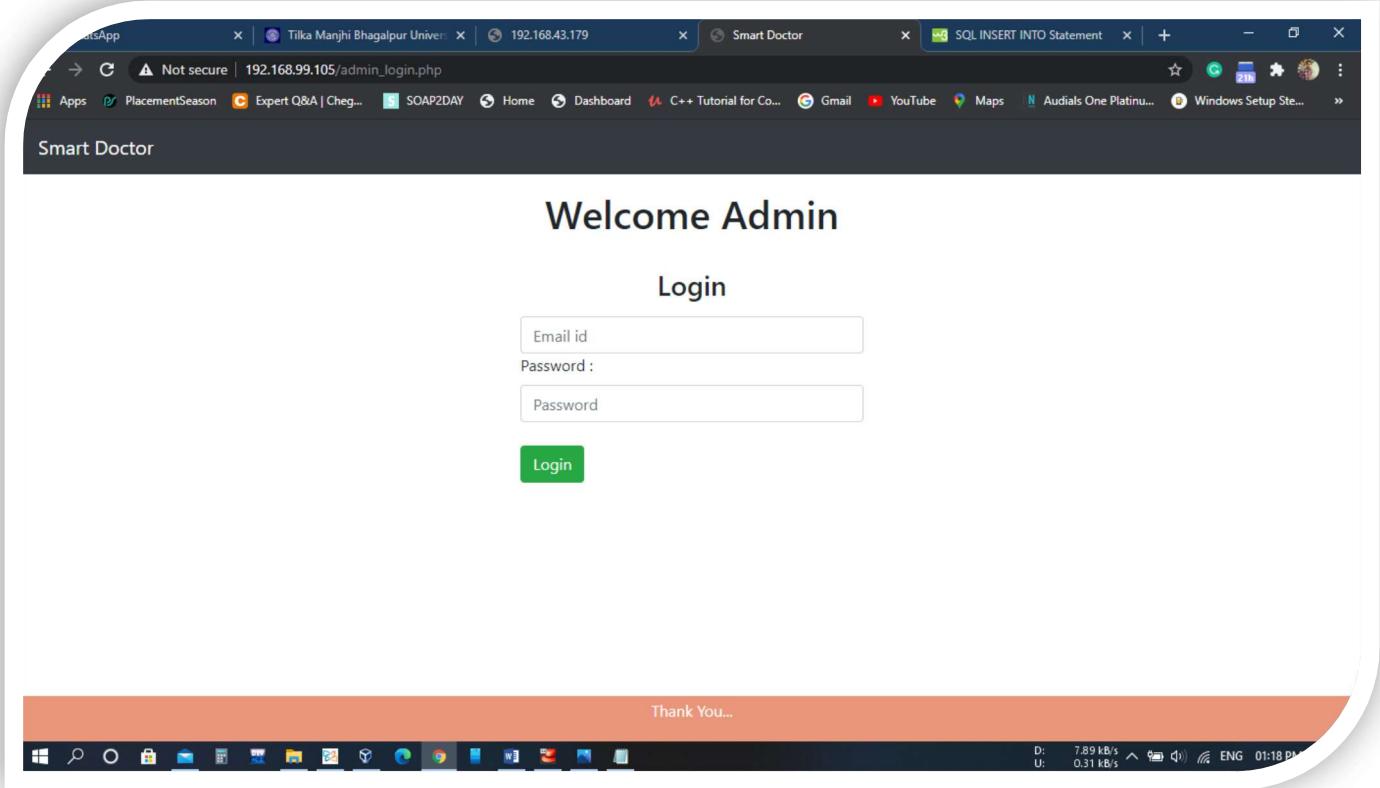
- * Location: The growths are most commonly found on the scalp, shoulder, chest, abdomen, or back. They are never found on the soles of the feet or palms of the hands.
- * Texture: Lesions usually start off as small, rough bumps, and they gradually get thicker and develop a warty surface. They can sometimes appear waxy and "stuck on."

Prediction output

Development of the website:-

The website is developed using html , javascript, css and php. It consist of login page , registration page, admin page, appointment booking facilities and user panel. Admin page or admin panel is operated by the admin to add the new doctors to the list , can see complete patient list. Website is powered with mysql as a information keeper. Php connect the website with the mysql.





Admin Login

Admin can login by entering the useremail and password. Admin has power to add any doctor remove doctor and control all website activity.

Patient List

S.No.	Name	Sex	Age	Address
1	gourav	M	8542	C/O-RAJIV KUNAR SINGH ,VIDHYAPURAM

Add Patient

Query From User

S.No.	Name	Email	Message
1	gkdfh	djghdgdfdfg	dfffsfd

Thank You...

Admin panel to view patient list and query

Smart Doctor Home

8	Dr. Shireen Y Poonwalla	shireenpoonwalla@gmail.com	Dermatologist	Shibani Poly Clinic 580, Opposite Bank of Maharashtra, Sachaipir Street,Camp, Pune – 411001
9	Dr. Aarohit Batra	aarohitbatra@gmail.com	Dermatologist	Derma World Skin Clinic,Q-4, Rajgouri Garden, near Janta Market Main EntryRajgouri Garden, West Delhi (Delhi).-110027
10	Dr. Amit Dutta	amitdutta@gmail.com	Dermatologist	Dr. Dutta's skin and Ayurvedic Treatment Centre,Jandiala-Jalandhar,Jalandhar (Punjab)-144003
11	Dr. Amit Gupta	amitguptapunjab@gmail.com	Dermatologist	Modern Hospital, Sutehri Road,Hoshiarpur (Punjab)-146001
12	Dr. Sooneita Wagh Markan	sooneitawaghdelhi@gmail.com	Dermatologist	Jail Rd, Pocket AL, Hari Nagar, New Delhi, Delhi 110064

[Add Doctor](#)

Appointment List

S.No.	Name	Email	Day	Time	Doctor Name	Message
1	efedf	gourav842k@gmail.com	Mon	am	Dr. Amonkar Krupa	dssdsd
2	gourav kumar	gouravk842@gmail.com	Wed	pm	Dr. Amonkar Krupa	erwrdf

D: 0.00 kB/s U: 0.00 kB/s ENG 01:19 PM

Appointment list, add doctor to maintain doctors

Smart Doctor Home

Doctor List

S.No.	Name	Email	Specialist	Address
1	Dr. Amonkar Krupa	amonkar123@gmail.com	Dermatologists	Trisha Polyclinic , Shri Sai Krishna Niwas , Shop no. 1, Agarbazar , S.K.Bole Road, Dadar (W), Dadar (West), Mumbai, Maharashtra – 400028
2	Dr. Abhishek De	abhishekde23@gmail.com	Dermatologist	Mani Square Mall, IT Building, 7th floor , Maniktala Main Road ,Kakurgachi,Kolkata – 700054
3	Dr. Aarti Sarda	aartisarda@gmail.com	Dermatologist	Mani Square Mall, IT Building, 7th floor , Maniktala Main Road ,Kakurgachi, Kolkata – 700054
4	Dr. Sachin Verma	sachinverma@gmail.com	Dermatologist	1, Apollo Gleneagles Hospital , Opposite Hyatt Residency,Salt Lake City – Bidhannagar . Kolkata-700064
5	Dr. Dhepes	dhepespune@gmail.com	Dermatologist	Runwal Regency Sadhu Vaswani Chowk, Above Gold Mart jewellers, Camp.Pune-411001
6	Dr. Batras Positive Health Clinic	batrasclinic@gmail.com	Dermatologist	Plot no. 2, 2nd floor, Akshay Complex, Pushpak Park, Opposite Sulzar India, Above FAB India, ITI Road, Aundh,Pune- 411007
7	Dr. Tarabai Limaye Hospital	tarabailimaye@gmail.com	Dermatologist	366, Narayan Peth, Opposite Gokhale Hall,Laxmi Road, Narayan Peth, Pune – 411030

D: 0.00 kB/s U: 0.00 kB/s ENG 01:19 PM

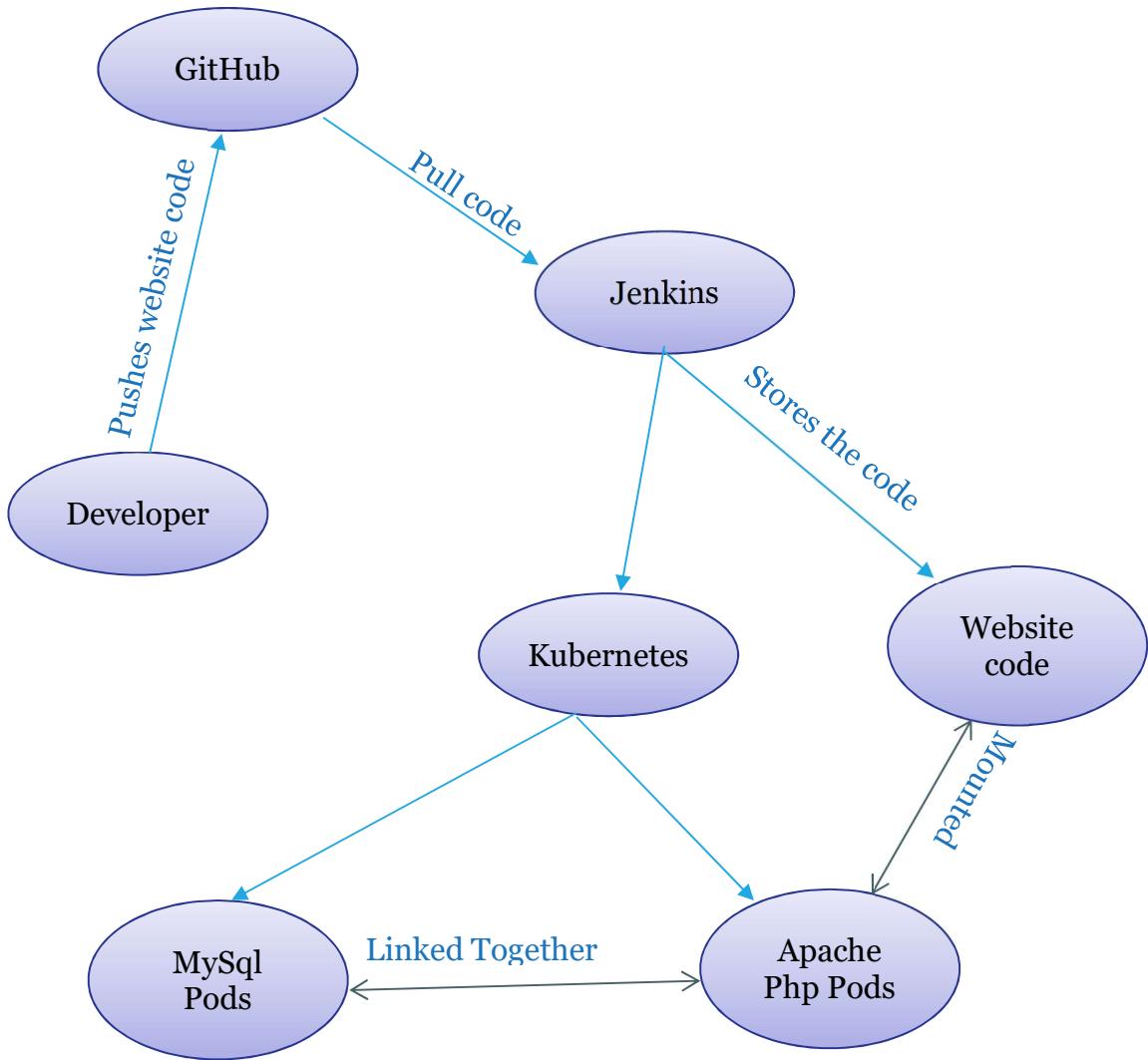
Doctor List

Deployment of the website:-

Website is deployed on the top of the kubernetes using the LAMP server(linux, Apache , MYSQL and PHP). With the use of the kubernetes smartdoctor website has achieved the power of the auto scaling. Auto scaling is technique used by the containers to balance the load on the server , i.e. whenever load increases on the website , kubernetes automatically launches new containers and redirect the traffic on the others preventing the server from crashing. The website also uses PVC (persistance volume claim) function of the kubernetes to keep the data permanent.

The complete deployment process is done in two steps,

1. Jenkins pull the latest website code from the github and store in the folder
2. Jenkins again triggers kubernetes to launch the website using the yml script

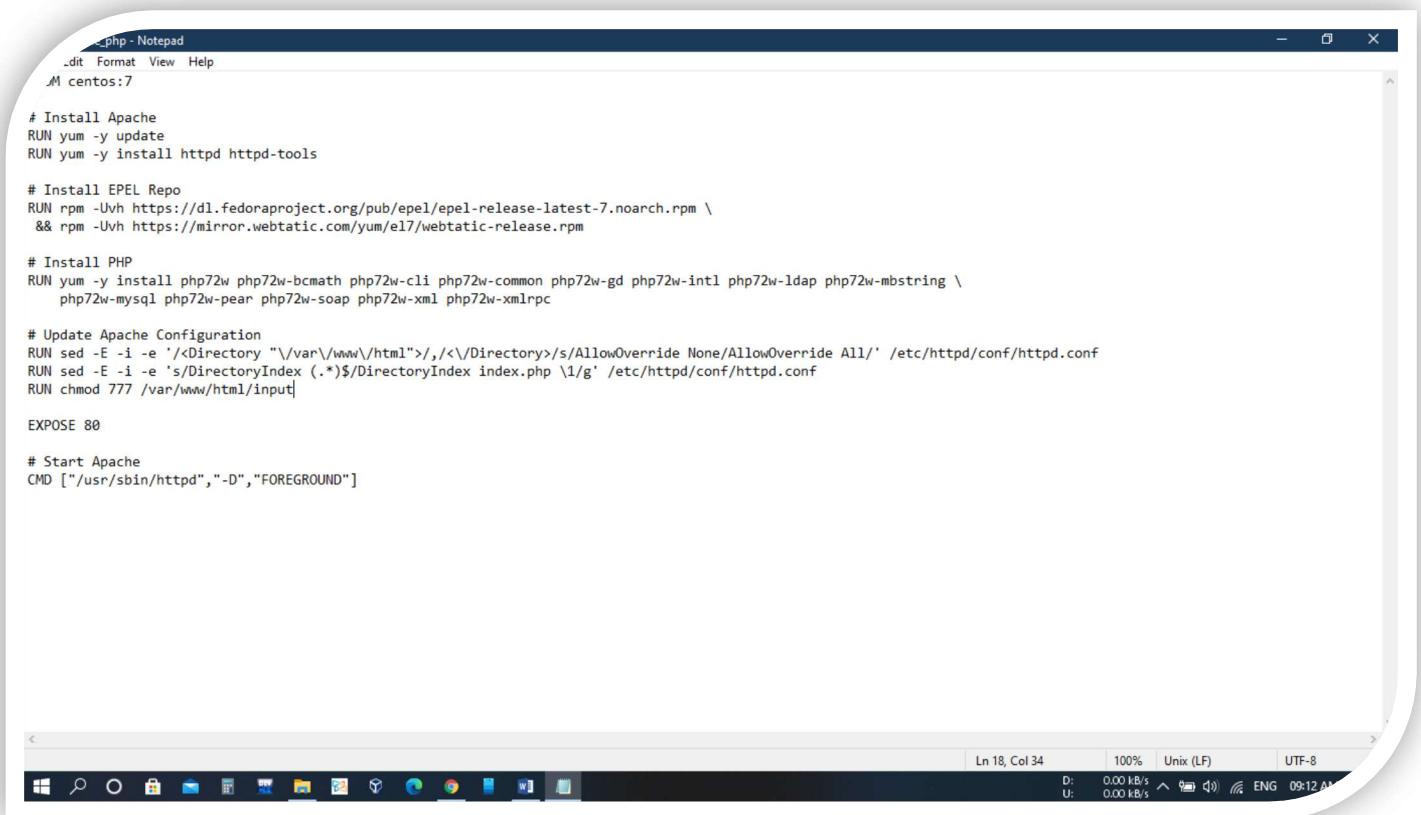


Creation of Docker Image

For this project three different docker image is created and pushed on the docker hub. A Docker image is the set of processes outlined in the Docker file. It is helpful to think of these as templates created by the Docker files. These are arranged in layers automatically. Each layer is dependent on the layer below it. Docker files are configuration files that “tell” Docker images what to install, update, etc. The Docker file says what to build that will be the Docker image. All the docker image is created on the using the dockerfile, especially to meet the needs of this project.

The following is the list of the created docker image

- manissha66/php-smartdoctor:new - This docker image is created to setup the apache and php server. This image has a feature of automatically start the server whenever the container is created using this image. The following is the snapshot of the dockerfile that is used to create this image.



```
#_php - Notepad
Edit Format View Help
M centos:7

# Install Apache
RUN yum -y update
RUN yum -y install httpd httpd-tools

# Install EPEL Repo
RUN rpm -Uvh https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm \
&& rpm -Uvh https://mirror.webtatic.com/yum/e17/webtatic-release.rpm

# Install PHP
RUN yum -y install php72w php72w-bcmath php72w-cli php72w-common php72w-gd php72w-intl php72w-ldap php72w-mbstring \
php72w-mysql php72w-pear php72w-soap php72w-xml php72w-xmlrpc

# Update Apache Configuration
RUN sed -E -i -e '/<Directory "\/var\/www\/html">/,<Directory>/s/AllowOverride None/AllowOverride All/' /etc/httpd/conf/httpd.conf
RUN sed -E -i -e 's/DirectoryIndex (.*)$/DirectoryIndex index.php \1/g' /etc/httpd/conf/httpd.conf
RUN chmod 777 /var/www/html/input

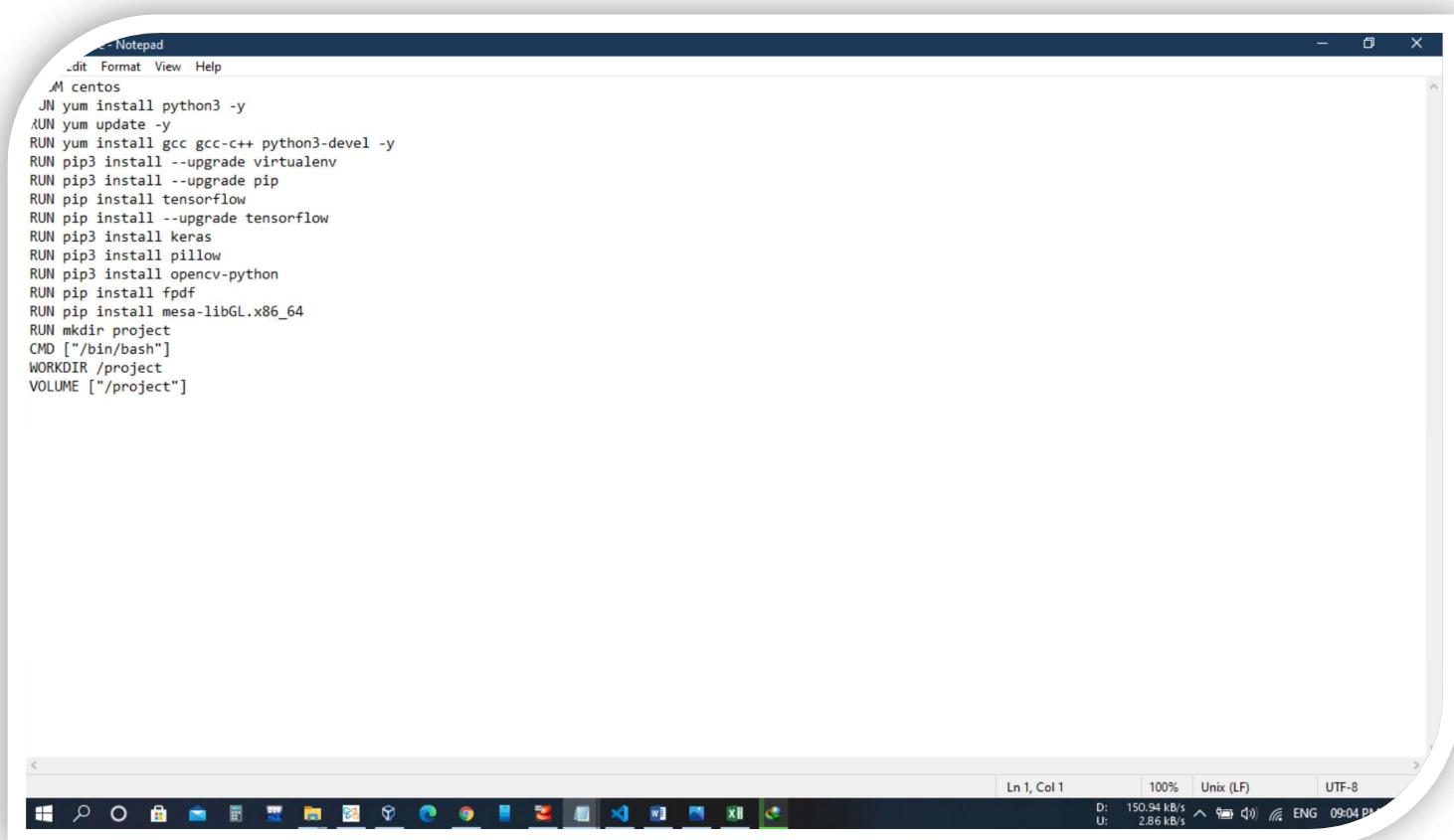
EXPOSE 80

# Start Apache
CMD ["/usr/sbin/httpd","-D","FOREGROUND"]
```

Apache Php DockerFile

- manissha66/training-prediction-image - This docker image trains the model with the given dataset and predict the output with the given input. It consists of the python module such as keras, tensorflow, opencv and fpdf.

The following is the snapshot of the dockerfile which is used to create this image.



```
-- Notepad
edit Format View Help
M centos
RUN yum install python3 -y
RUN yum update -y
RUN yum install gcc gcc-c++ python3-devel -y
RUN pip3 install --upgrade virtualenv
RUN pip3 install --upgrade pip
RUN pip install tensorflow
RUN pip install --upgrade tensorflow
RUN pip3 install keras
RUN pip3 install pillow
RUN pip3 install opencv-python
RUN pip install fpdf
RUN pip install mesa-libGL.x86_64
RUN mkdir project
CMD ["/bin/bash"]
WORKDIR /project
VOLUME ["/project"]
```

Dockerfile for training prediction image

- manissha66/mysql-php:new - This image is launches the mysql server with the pre created all the necessary databases, tables and have the default value inserted into it.



```

Dockerfile_mysql - Notepad
File Edit Format View Help
FROM mysql:5.7 as builder

# That file does the DB initialization but also runs mysql daemon, by removing the last line it will only init
RUN ["sed", "-i", "s/exec \"$@\"/echo \"not running $@\\"/", "/usr/local/bin/docker-entrypoint.sh"]

# needed for initialization
ENV MYSQL_ROOT_PASSWORD=root

COPY setup.sql /docker-entrypoint-initdb.d

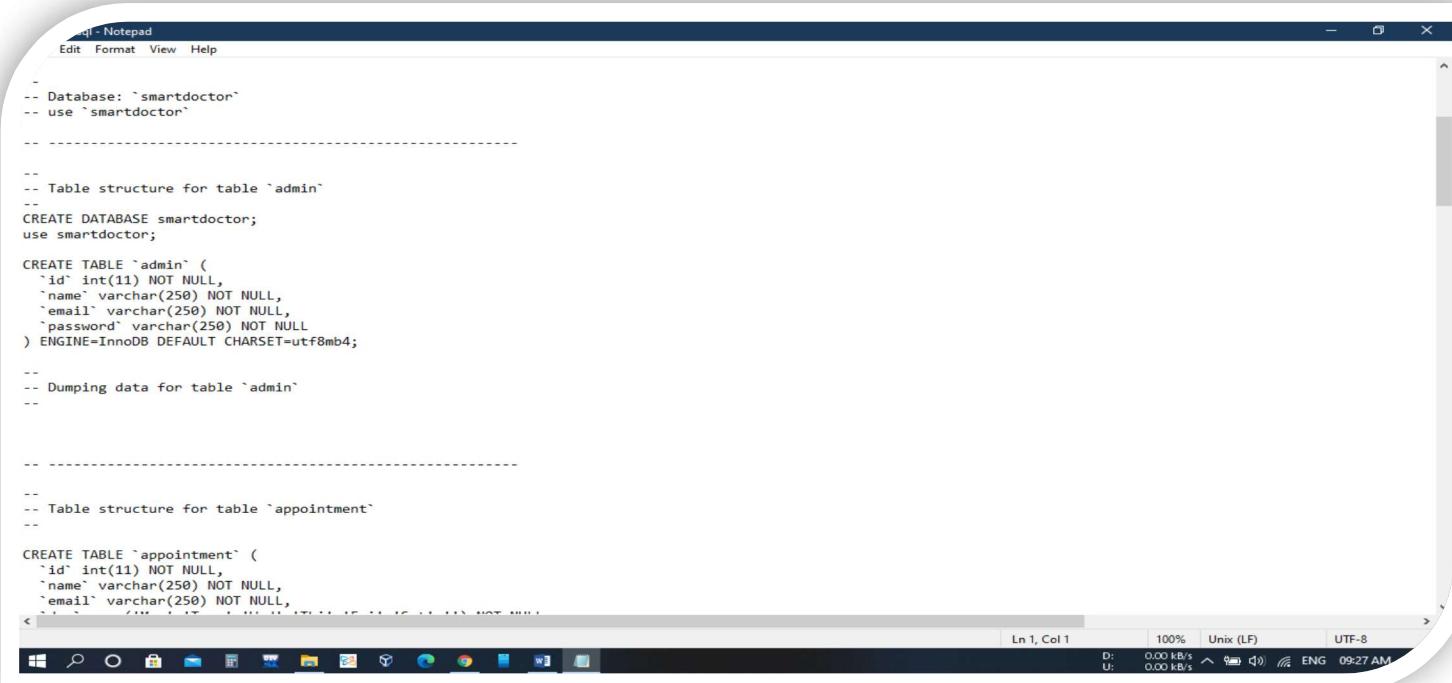
# Need to change the datadir to something else that /var/lib/mysql because the parent docker file defines it as a volume.
# https://docs.docker.com/engine/reference/builder/#volume :
#     Changing the volume from within the Dockerfile: If any build steps change the data within the volume after
#     it has been declared, those changes will be discarded.
RUN ["/usr/local/bin/docker-entrypoint.sh", "mysqld", "--datadir", "/initialized-db"]

FROM mysql:5.7

COPY --from=builder /initialized-db /var/lib/mysql

```

Dockerfile of mysql image



```

-- Database: `smartdoctor`
-- use `smartdoctor`

---

-- Table structure for table `admin`
-- CREATE DATABASE smartdoctor;
use smartdoctor;

CREATE TABLE `admin` (
  `id` int(11) NOT NULL,
  `name` varchar(250) NOT NULL,
  `email` varchar(250) NOT NULL,
  `password` varchar(250) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

-- Dumping data for table `admin`
--


---

-- Table structure for table `appointment`
-- CREATE TABLE `appointment` (
  `id` int(11) NOT NULL,
  `name` varchar(250) NOT NULL,
  `email` varchar(250) NOT NULL,
  `date` date NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

```

SQL script

Jenkins Setup

Jenkins is a free and open source automation server. It helps automate the parts of software development related to building, testing, and deploying, facilitating continuous integration and continuous delivery. It is a server-based system that runs in servlet containers such as Apache Tomcat. SmartDoctor uses Jenkins efficiently to integrate all the technologies and auto the things. No manual work is required in this project, everything is automatic.

The screenshot shows the Jenkins Dashboard with a list of jobs. The jobs are:

W	Name	Last Success	Last Failure	Last Duration
Cloud	code-pulling-for-train-the-model	1 day 13 hr - #20	18 hr - #21	40 min
Sun	code-pulling-job-for-websiteHosting	1 hr 24 min - #43	13 hr - #35	22 sec
Cloud with rain	model-training-job	2 mo 10 days - #15	11 min - #42	5 min 52 sec
Sun with rain	training-code-testing-job	11 min - #31	18 hr - #27	0.31 sec
Cloud with rain	uploading-job	19 min - #35	33 min - #33	17 sec
Sun	websiteHosting-job	45 min - #25	N/A	1.9 sec

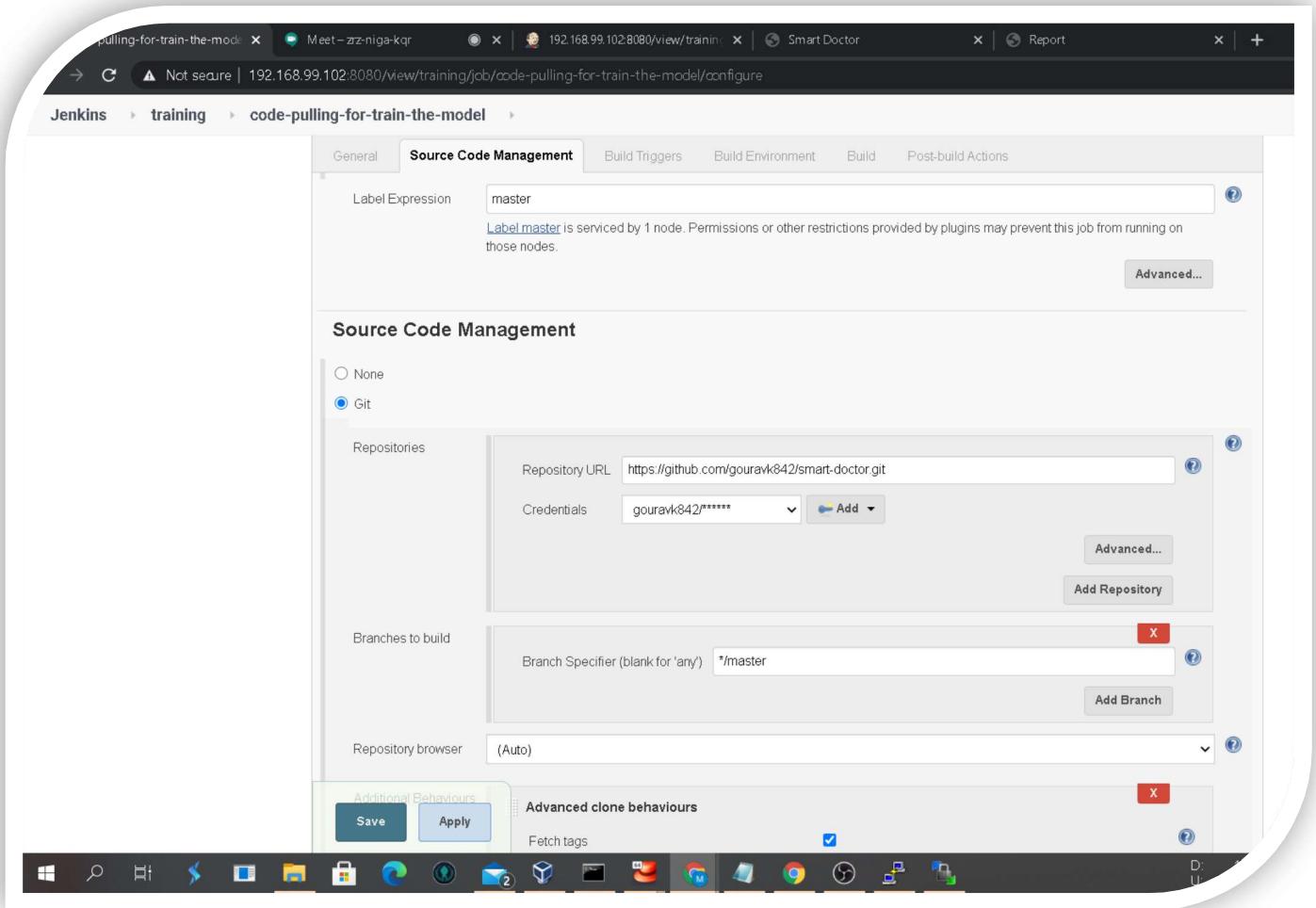
Jenkins DashBoard

As clearly seen in the above picture, a set of Jenkins job is created. All these job does different type of the integrations.

Job Name	Working
Code-pulling-for-training-the-model	This job download the dataset from the github
Code-pulling-job-for-websiteHosting	This job pulls the latest website code from the github
Model-training-job	This job trains and creates the model with the latest dataset
Uploading-job	This job maintains the prediction part
websiteHosting-job	This job host the website on the top of kubernetes

Code-pulling-for-training-the-model – The work of this job is to keep the model up to date. This job is triggered whenever the new dataset is pushed on the github. This job download all the dataset from the github and stores in the desired location. After the successful completion of this job, it triggers the second job named ‘model-training job’.

The following is the snapshot of this job showing its complete setting



The screenshot shows the Jenkins configuration page for the 'code-pulling-for-train-the-model' job. The 'Build Triggers' tab is selected. Under 'Poll SCM', the schedule is set to '*****'. A warning message states: 'Do you really mean "every minute" when you say "*****"? Perhaps you meant "H * * * * to poll once per hour'. The 'Build Environment' tab is also visible, showing options like 'Delete workspace before build starts' and 'Add timestamps to the Console Output'. At the bottom, there are 'Save' and 'Apply' buttons.

The screenshot shows the Jenkins configuration page for the 'code-pulling-for-train-the-model' job. The 'Build' tab is selected, showing a single 'Execute shell' step with the command: 'sudo cp -rvf * /root/github_code/'. Below this, the 'Post-build Actions' section is shown, with an 'Add post-build action' dropdown menu. At the bottom, there are 'Save' and 'Apply' buttons.

Code-pulling-job-for-websiteHosting – This job is triggered whenever new website code is uploaded on the github. This job download the new website code from the github and store it in the given location. This job triggers the next job ‘websiteHosting Job’.

The following is the snapshot of this job.

Jenkins Console Output for job #43:

```
Started by user admin
Running as SYSTEM
Building on master in workspace /var/lib/jenkins/workspace/code-pulling-job-for-websiteHosting
No credentials specified
> git rev-parse --is-inside-work-tree # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/gouravk842/smarddoctorwebsite.git # timeout=10
Fetching upstream changes from https://github.com/gouravk842/smarddoctorwebsite.git
> git -version # timeout=10
> git fetch --tags --progress -- https://github.com/gouravk842/smarddoctorwebsite.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/master^{commit} # timeout=10
> git rev-parse refs/remotes/origin/origin/master^{commit} # timeout=10
Checking out Revision cf4b1ce6c76cf160a49dd1baf795223ac80abc (refs/remotes/origin/master)
> git config core.sparsecheckout # timeout=10
> git checkout cf4b1ce6c76cf160a49dd1baf795223ac80abc # timeout=10
Commit message: "Update connectToDatabase.php"
> git rev-list --no-walk cf4b1ce6c76cf160a49dd1baf795223ac80abc # timeout=10
[code-pulling-job-for-websiteHosting] $ /bin/sh -xe /tmp/jenkins201501054791123307.sh
+ sudo cp -rnf addpatient.php admin_Login.php admin.php connectToDatabase.php css delete_doctor.php doctorAddPatient.php doctor.php fonts helpQuery.php
images index.php input insert_file.php js login.php logout.php output patient.php patientRegistration.php showPatient.php style.css /root/smard_doctor/
'addpatient.php' -> '/root/smard_doctor/addpatient.php'
'admin_login.php' -> '/root/smard_doctor/admin_login.php'
'admin.php' -> '/root/smard_doctor/admin.php'
'connectToDatabase.php' -> '/root/smard_doctor/connectToDatabase.php'
'css/admin.css' -> '/root/smard_doctor/css/admin.css'
'css/animate.css' -> '/root/smard_doctor/css/animate.css'
'css/bootstrap-theme.css' -> '/root/smard_doctor/css/bootstrap-theme.css'
'css/bootstrap-theme.css.map' -> '/root/smard_doctor/css/bootstrap-theme.css.map'
'css/bootstrap-theme.min.css' -> '/root/smard_doctor/css/bootstrap-theme.min.css'
'css/bootstrap-theme.min.css.map' -> '/root/smard_doctor/css/bootstrap-theme.min.css.map'
'css/bootstrap.css' -> '/root/smard_doctor/css/bootstrap.css'
'css/bootstrap.css.map' -> '/root/smard_doctor/css/bootstrap.css.map'
'css/admin.css' -> '/root/smard_doctor/css/admin.css'
'css/animate.css' -> '/root/smard_doctor/css/animate.css'
'css/bootstrap-theme.css' -> '/root/smard_doctor/css/bootstrap-theme.css'
'css/bootstrap-theme.css.map' -> '/root/smard_doctor/css/bootstrap-theme.css.map'
'css/bootstrap-theme.min.css' -> '/root/smard_doctor/css/bootstrap-theme.min.css'
'css/bootstrap-theme.min.css.map' -> '/root/smard_doctor/css/bootstrap-theme.min.css.map'
'css/bootstrap.css' -> '/root/smard_doctor/css/bootstrap.css'
'css/bootstrap.css.map' -> '/root/smard_doctor/css/bootstrap.css.map'
```

Jenkins Console Output for job #43:

```
'images/prettyphoto/dark_rounded/btnPrevious.png' -> '/root/smard_doctor/images/prettyphoto/dark_rounded/btnPrevious.png'
'images/prettyphoto/dark_rounded/contentPattern.png' -> '/root/smard_doctor/images/prettyphoto/dark_rounded/contentPattern.png'
'images/prettyphoto/dark_rounded/loader.gif' -> '/root/smard_doctor/images/prettyphoto/dark_rounded/loader.gif'
'images/prettyphoto/default_thumbnail.gif' -> '/root/smard_doctor/images/prettyphoto/default_thumbnail.gif'
'images/prettyphoto/dark_rounded/loader.gif' -> '/root/smard_doctor/images/prettyphoto/dark_rounded/loader.gif'
'images/prettyphoto/light_rounded/loaderNext.png' -> '/root/smard_doctor/images/prettyphoto/light_rounded/loaderNext.png'
'images/prettyphoto/light_rounded/loaderPrevious.png' -> '/root/smard_doctor/images/prettyphoto/light_rounded/loaderPrevious.png'
'images/prettyphoto/light_rounded/defaultThumbnail.gif' -> '/root/smard_doctor/images/prettyphoto/light_rounded/defaultThumbnail.gif'
'images/prettyphoto/light_rounded/loader.gif' -> '/root/smard_doctor/images/prettyphoto/light_rounded/loader.gif'
'images/prettyphoto/light_rounded/loaderNext.png' -> '/root/smard_doctor/images/prettyphoto/light_rounded/loaderNext.png'
'index.php' -> '/root/smard_doctor/index.php'
'input/WhatsApp Image 2020-09-17 at 11.37.22 AM.jpeg' -> '/root/smard_doctor/input/WhatsApp Image 2020-09-17 at 11.37.22 AM.jpeg'
'input/file.php' -> '/root/smard_doctor/input/file.php'
'js/all.js' -> '/root/smard_doctor/js/all.js'
'js/animate.js' -> '/root/smard_doctor/js/animate.js'
'js/custom.js' -> '/root/smard_doctor/js/custom.js'
'js/navmenu.js' -> '/root/smard_doctor/js/navmenu.js'
'js/prettyphoto.js' -> '/root/smard_doctor/js/jquery.prettyPhoto.js'
'js/jquery.vide.js' -> '/root/smard_doctor/js/jquery.vide.js'
'js/map.js' -> '/root/smard_doctor/js/map.js'
'js/modernizer.js' -> '/root/smard_doctor/js/modernizer.js'
'js/owl.carousel.js' -> '/root/smard_doctor/js/owl.carousel.js'
'js/portfolio.js' -> '/root/smard_doctor/js/portfolio.js'
'js/retilna.js' -> '/root/smard_doctor/js/retilna.js'
'js/slick.js' -> '/root/smard_doctor/js/slick.js'
'login.php' -> '/root/smard_doctor/login.php'
'logout.php' -> '/root/smard_doctor/logout.php'
'output/export.pdf' -> '/root/smard_doctor/output/export.pdf'
'patient.php' -> '/root/smard_doctor/patient.php'
'patientRegistration.php' -> '/root/smard_doctor/patientRegistration.php'
'showPatient.php' -> '/root/smard_doctor/showPatient.php'
'styles.css' -> '/root/smard_doctor/styles.css'
[code-pulling-job-for-websiteHosting] $ /bin/sh -xe /tmp/jenkins2273160569742536912.sh
+ sudo scp -r /root/smard_doctor docker@192.168.99.106:/home/docker/
Triggering a new build of websiteHosting:job
```

Log of the job

Model-training-job – This job is triggered by the job ‘code-pulling-for-train-the-model’. This job launch the docker container and trains the model with the new dataset. This job is designed in such a way that it automatically increases the no of epoch if the accuracy of the model is not 80%. It stops after the model achieved 80% accuracy.

The screenshot shows the Jenkins job configuration interface for a job named 'model-training-job'. The 'Build' tab is selected. Under the 'Execute shell' step, there is a command block containing the following Docker script:

```
x=1
sudo cd /root/github_code
sudo docker start training
sudo docker exec training python3 /root/skin_cancer.py $x
accuracy=$(sudo docker exec training cat accuracy.txt)
while [ $accuracy -lt 70 ]
do
if sudo docker ps -a | grep training
then
echo "already running"
sudo docker start training
let 'x+=1'
sudo docker exec training python3 /root/skin_cancer.py $x
accuracy=$(sudo docker exec predictor1 cat accuracy.txt)
else
sudo docker run -dit --name training -v /root/github_code:/root/manissha66/training-prediction-image
fi
done

cp -rvf /root/github_code/trained.h5 /root/smartdoctor_prediction
```

Below the command block, there is a link to "See the list of available environment variables". At the bottom of the build step, there are "Save" and "Apply" buttons.

Configuration of Jenkins job

```

Jenkins > training > model-training-job > #29
[...]
keras
+ echo 'already running'
already running
+ sudo docker start kerasos
kerasos
+ let x=1
+ sudo docker exec kerasos python3 skin_cancer.py 106
2020-09-21 03:13:49.009478: W tensorflow/stream_executor/platform/default/dso_loader.cc:59] Could not load dynamic library 'libcudart.so.10.1'; dlerror: libcudart.so.10.1: cannot open shared object file: No such file or directory
2020-09-21 03:13:49.009525: I tensorflow/stream_executor/cuda/cudart_stub.cc:29] Ignore above cudart dlerror if you do not have a GPU set up on your machine.
2020-09-21 03:13:52.241510: W tensorflow/stream_executor/platform/default/dso_loader.cc:59] Could not load dynamic library 'libcuda.so.1'; dlerror: libcuda.so.1: cannot open shared object file: No such file or directory
2020-09-21 03:13:52.241580: W tensorflow/stream_executor/cuda/cuda_driver.cc:312] failed call to cuInit: UNKNOWN ERROR (303)
2020-09-21 03:13:52.241588: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:156] kernel driver does not appear to be running on this host (b3a5fbfe3eb01). /proc/driver/nvidia/version does not exist
2020-09-21 03:13:52.242202: I tensorflow/core/platform/cpu_feature_guard.cc:142] This TensorFlow binary is optimized with oneAPI Deep Neural Network Library (oneDNN) to use the following CPU instructions in performance-critical operations: AVX2
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
2020-09-21 03:13:52.259267: W tensorflow/core/platform/profile_utils/cpu_utils.cc:184] CPU Frequency: 240000000 Hz
2020-09-21 03:13:52.259748: I tensorflow/compiler/xla/service/service.cc:168] XLA service 0x55d3a0e0670 initialized for platform Host (this does not guarantee that XLA will be used). Devices:
2020-09-21 03:13:52.259783: I tensorflow/compiler/xla/service/service.cc:176] StreamExecutor device (0): Host, Default Version
0 InputLayer False
1 ZeroPadding2D False
2 Conv2D False
3 BatchNormalization False
4 ReLU False
5 DepthwiseConv2D False
6 BatchNormalization False
7 ReLU False
8 Conv2D False
9 BatchNormalization False
10 Relu False
11 DepthwiseConv2D False
12 DepthwiseConv2D False
13 BatchNormalization False
14 ReLU False
15 Conv2D False
16 BatchNormalization False
[...]

```

```

Jenkins > training > model-training-job > #29
[...]
conv_pw_13_bn (BatchNormaliz (None, 1, 1, 1024) 4096
conv_pw_13_relu (ReLU) (None, 7, 7, 1024) 0
global_average_pooling2d (G (None, 1024) 0
dense (Dense) (None, 1024) 1049600
dense_1 (Dense) (None, 1024) 1049600
dense_2 (Dense) (None, 512) 524800
dense_3 (Dense) (None, 9) 4617
=====
Total params: 5,857,481
Trainable params: 2,628,617
Non-trainable params: 3,228,864
[...]
current accuracy: 29.968735575675964
++ sudo docker exec kerasos cat accuracy.txt
+ accuracy=29
+ '[' 29 -lt 70 ']'
+ sudo docker ps -a
+ grep kerasos
b3a5fbfe3eb01      manishsha66/image-tensorflow-keras:new    "/bin/bash"          18 hours ago   Up 10 hours
kerasos
+ echo 'already running'
already running
+ sudo docker start kerasos
kerasos
+ let x=1
+ sudo docker exec kerasos python3 skin_cancer.py 107
2020-09-21 03:17:30.935998: W tensorflow/stream_executor/platform/default/dso_loader.cc:59] Could not load dynamic library 'libcudart.so.10.1'; dlerror: libcudart.so.10.1: cannot open shared object file: No such file or directory
2020-09-21 03:17:30.936047: I tensorflow/stream_executor/cuda/cudart_stub.cc:29] Ignore above cudart dlerror if you do not have a GPU set up on your machine.
2020-09-21 03:17:34.299889: W tensorflow/stream_executor/platform/default/dso_loader.cc:59] Could not load dynamic library 'libcuds.so.1'; dlerror: libcuds.so.1: cannot open shared object file: No such file or directory
2020-09-21 03:17:34.299943: W tensorflow/stream_executor/cuda/cuda_driver.cc:312] failed call to cuInit: UNKNOWN ERROR (303)
2020-09-21 03:17:34.299966: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:156] kernel driver does not appear to be running on this host (b3a5fbfe3eb01). /proc/driver/nvidia/version does not exist
[...]

```

The above picture shows that first time the accuracy was not upto the mark, hence again restart the training with the increased no of the epoch.

Uploading-job – This job is triggered just after the user uploads the image in the website. The purpose of this job is to get the image uploaded by the user, start the docker container , run the prediction python script, and copy back the output file to the website location which can be provided to the user as an output.

The screenshot shows the Jenkins job configuration for 'uploading-job'. It has two build steps:

- Execute shell script on remote host using ssh**: SSH site is set to 'docker@192.168.99.106:22'. The command is: `scp -r /home/docker/smardocor/input root@192.168.99.102:/root/smardocor_prediction/`
- Execute shell**: Command is a shell script to start a Docker container named 'predictorCon' and run a Python script inside it.

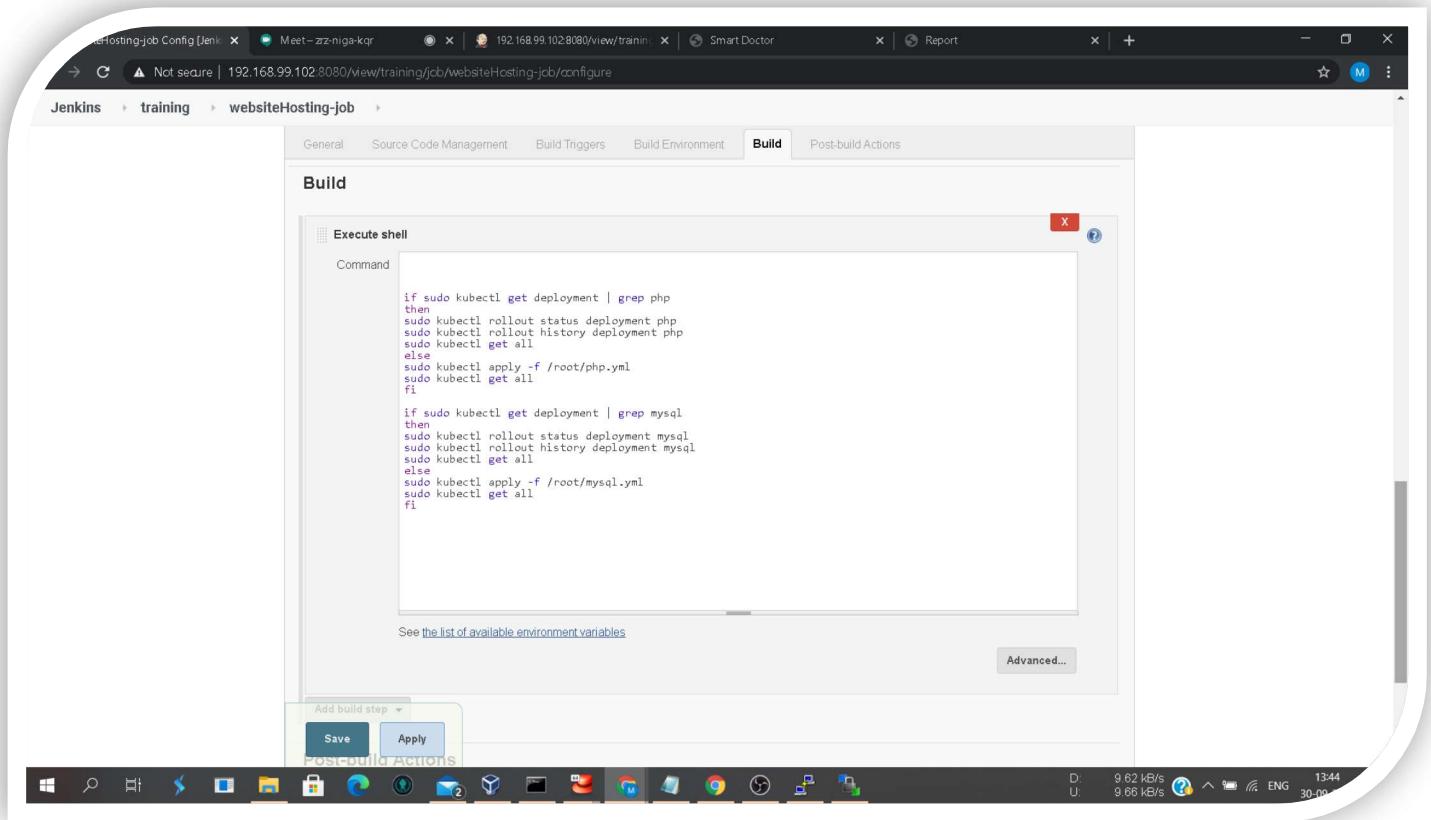
Buttons at the bottom include 'Save', 'Apply', and 'ADVANCED'.

Configuration of Jenkins job

```
[SSH] executing...
[SSH] completed
[SSH] exit-status: 0
[uploading-job] $ /bin/sh -xe /tmp/jenkins160965155889094128.sh
+ sudo docker ps -a
+ grep predictorCon
dbdf7a08a7de    predictorCon   "/bin/bash"        14 hours ago   Up 2 hours
+ echo "container is in running state"
+ echo "container is in running state"
+ sudo docker start predictorCon
predictorCon
+ sudo docker exec predictorCon python3 /root/predictor.py
2020-09-30 07:58:04.988202: W tensorflow/stream_executor/platform/default/dso_loader.cc:59] Could not load dynamic library 'libcudart.so.10.1'; dlerror: libcudart.so.10.1: cannot open shared object file: No such file or directory
2020-09-30 07:58:04.988504: I tensorflow/stream_executor/cuda/cuda_stub.cc:29] Ignore above cudart dlerror if you do not have a GPU set up on your machine.
2020-09-30 07:58:11.211054: W tensorflow/stream_executor/platform/default/dso_loader.cc:59] Could not load dynamic library 'libcuda.so.1'; dlerror: libcuda.so.1: cannot open shared object file: No such file or directory
2020-09-30 07:58:11.211054: W tensorflow/stream_executor/cuda/cuda_driver.cc:312] failed call to cuInit: UNKNOWN ERROR (993)
2020-09-30 07:58:11.211098: I tensorflow/stream_executor/cuda/cuda_diagnostic.cc:156] kernel driver does not appear to be running on this host (cuda).
2020-09-30 07:58:11.212446: I tensorflow/core/platform/cuda_feature_guard.cc:142] This TensorFlow binary is optimized with oneAPI Deep Neural Network Library (oneDNN) and will use the following CPU optimizations in performance-critical operations: AVX2
2020-09-30 07:58:11.242076: I tensorflow/core/platform/profile_utils/cpu_utils.cc:184] CPU Frequency: 2399995000 Hz
2020-09-30 07:58:11.243021: I tensorflow/core/platform/xla/service/service.cc:108] XLA service 0x55ad6ade1550 initialized for platform Host (this does not guarantee that XLA will be used). Devices:
2020-09-30 07:58:11.243066: I tensorflow/compiler/xla/service/service.cc:176] StreamExecutor device 0: Host, Default Version
[uploading-job] $ /bin/sh -xe /tmp/jenkins160965155889094128.sh
+ sudo scp -r /root/smardocor_prediction/output/output.pdf docker@192.168.99.106:/home/docker/smardocor/output/
finished: SUCCESS
```

Page generated Sep 30, 2020 1:46:12 PM IST REST API Jenkins ver. 2.222.3

websiteHosting job – This job is triggered by job ‘code-pulling-job-for-websiteHosting’. The work of this job is to launch the website on the kubernetes using the yml script. Yml script is the kubernetes script which contains all the information of container names, service, persistence volume for the kubernetes pods.



Configuration of Jenkins job

The screenshot shows a Windows desktop environment with a Jenkins job console output window open. The title bar of the window reads "Console Output". The main content of the window displays the command-line logs of a Jenkins job named "websiteHosting-job #25". The logs show the execution of several commands, including `bin/sh -xe /tmp/jenkins5925621298109572346.sh`, `sudo kubectl get deployment`, `grep php`, `sudo kubectl apply -f /root/php.yml`, `service/php created`, `persistentvolumeclaim/php-pv-claim unchanged`, `deployment.apps/php created`, `sudo kubectl get all`, and `grep mysql` followed by similar commands for MySQL. The logs also show the creation of Kubernetes resources like services and replicaset. The Jenkins interface includes a sidebar with links like "Back to Project", "Status", "Changes", "Console Output" (which is selected), "View as plain text", "Edit Build Information", "Delete build #25", and "Previous Build". The bottom of the window shows the Jenkins navigation bar with links for "Jenkins", "training", "websiteHosting-job", and "#25". The desktop taskbar at the bottom has icons for various applications like File Explorer, Edge, and FileZilla. The system tray shows network speed (D: 11.53 KB/s, U: 17.61 KB/s), battery level (30%), and system status (ENG). The system clock in the top right corner shows 13:46.

```
Started by user admin
Running as SYSTEM
Building on master in workspace /var/lib/jenkins/workspace/websiteHosting-job
[websiteHosting-job] $ /bin/sh -xe /tmp/jenkins5925621298109572346.sh
+ sudo kubectl get deployment
+ grep php
No resources found in default namespace.
+ sudo kubectl apply -f /root/php.yml
service/php created
persistentvolumeclaim/php-pv-claim unchanged
deployment.apps/php created
+ sudo kubectl get all
NAME                      READY   STATUS    RESTARTS   AGE
pod/php-54f7c646ff-zbv2b  0/1     ContainerCreating   0          0s

NAME              TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)          AGE
service/kubernetes  ClusterIP   10.96.0.1      <none>        443/TCP        32s
service/php        LoadBalancer  10.105.156.92  <pending>    80:31565/TCP  0s

NAME          READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/php  0/1       1           0          0s

NAME          DESIRED   CURRENT   READY   AGE
replicaset.apps/php-54f7c646ff  1         1         0         0s
+ sudo kubectl get deployment
+ grep mysql
+ sudo kubectl apply -f /root/mysql.yml
service/mysql created
persistentvolumeclaim/mysql-pv-claim unchanged
deployment.apps/mysql created
+ sudo kubectl get all
NAME                      READY   STATUS    RESTARTS   AGE
pod/mysql-56445d446d-2qj6t  0/1     ContainerCreating   0          0s
```

Log of website hosting job

```
Notepad
File Edit Format View Help
Version: v1
kind: Service
metadata:
  name: php
  labels:
    app: php
spec:
  ports:
    - port: 80
      nodePort: 31565
      protocol: TCP
  selector:
    app: php
    tier: frontend
  type: LoadBalancer
---
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: php-pv-claim
  labels:
    app: php
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 20Gi
---
apiVersion: apps/v1 # for versions before 1.9.0 use apps/v1beta2
kind: Deployment
metadata:
  name: php
  labels:
    app: php
spec:
```

Yml code

Github Files

The following is the github url of all the code of this project.

<https://github.com/gouravk842/smart-doctor.git>

github repo consists of various folders and files. These are

- dataset – consist the dataset on which model is trained
- dockerfiles – consist of all the dockerfile which is used to create the docker image used in this project.
- Precautionary tips – contains all the medicine details of the diseases.
- Python scripts – contains python code used for training and prediction
- Website code – smartdoctor website code
- Yml code – kubernetes yml script used to launch the pods
- Setup.sql – sql file that contains details of required tables and rows
- Trained.h5 – trained model

Future Advancement

In future, the complete setup which is currently running on the local system can be shifted on aws cloud. This will make our website public and anyone can visit it and make use of it. The model which is trained on our personal laptop (low configuration laptop) can be shifted to cloud computing, the better the efficiency of the machine the faster will be training and prediction. Various new features will be added to it.

One thing we can proudly say, smartdoctor is a powerful setup , which will never become outdated in the market.